Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (currently amended) A method comprising

displaying a control in a graphical user interface;

receiving a data request[s] generated by an application executing on a first system;

generating an Extensible Markup Language (XML) structure for [~~each~~] the data request;

the XML structure including

a variable stream of data stored in a memory of the first system, the stream including

an XML element for [~~each~~] the request, the XML element including

data from a data set object;

transferring the XML structure to a second system;

receiving a response from the second system;

updating the data set object based on the response; and

[~~updating controls in a graphical user interface~~] changing the value of the control based on the updated data set object.

2. (Canceled).

3. (Canceled).

4. (Previously presented) The method of claim 1 in which the XML element is a class object whose data is stored to generate XML.

5. (Canceled).

6. (Previously presented) The method of claim 1 in which the data set object includes table dictionaries, column names and data from record sets, and stored procedure parameters.

7. (Previously presented) The method of claim 1, wherein transferring the XML structure comprises using a text transmission protocol.

8. (Previously presented) The method of claim 7, in which the text transmission protocol is Hypertext Transfer Protocol (HTTP).

9. (Previously presented) The method of claim 1, further comprising:

parsing the XML structure into request statements; and

executing the request statements.

10. (Original) The method of claim 9 further comprising:

translating responses from the executed request statements into an XML format; and

sending the XML formatted responses to the first system.

11. (Currently amended) A method for implementing a distributed application protocol, the method comprising:

displaying a control in a graphical user interface;

receiving an application request from an application in a first system;

translating the application request into a data structure, the data structure being a standardized delimited data structure stored in a memory of the first system, and

transforming the data structure into a stream of text based data utilizing an Extensible Markup Language (XML) format;

transmitting the stream of text to a second system over a network, transmitting causing the second system to execute an executable command;

receiving a response in the first system; and

[~~adjusting the graphical user interface~~] changing the value of the control based on the response.

12. (Canceled).

13. (Previously presented) The method of claim 11, further comprising :

causing the second system to parse the stream of text by breaking down the stream of text to an executable command format utilizing data and parameters related to an application.

14. (Previously presented) The method of claim 13, further comprising causing the second system to evaluate the executable command prior to execution in the second system.

15. (Previously presented) The method of claim 14, further comprising causing the second system to evaluate a result generated by executing the executable commands.

16. (Previously presented) The method of claim 15 further comprising:

causing the second system to convert a result into a stream of text based data in a standardized XML format; and

transmitting the result over the network to the first system.

17. (Currently amended) A method for implementing a distributed application protocol, the method comprising:

displaying a control in a graphical user interface;

generating a first data structure for storing data and parameters received from an application residing in the server, the first data structure including database tables, procedure results from logic calls and status/error messages;

translating application requests from the application into a delimited second data structure, stored in a memory, the second data structure having an element for each of the application requests, the application requests being generated in response to user actions in a graphical user interface; [~~and~~]

generating a stream of text-based data in an Extensible Markup Language (XML) format from the second data structure;

transmitting the stream;

receiving a response; and

changing the value of the control based on the response.

18. (Canceled).

19. (Canceled).

20. (Previously presented) The method of claim 17 in which the element is a class object.

21. (Previously presented) A method comprising:

in a server, receiving a stream of text-based data in an Extensible Markup Language (XML) format;

parsing the stream into request statements;

intercepting the request statements prior to execution; and

applying additional logic based on a type or content of the request.

22. (Canceled)

23. (Original) The method of claim 21 in which executing further comprises applying additional logic to responses generated from executing the request statements.

24. (Original) The method of claim 21 further comprising:

converting responses generated from each of the executed request statements into an XML format.

25. (Canceled).

26. (Currently amended) A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

display a control in a graphical user interface;

generate a first data structure for storing data and parameters related to an application residing in the server, the first data structure comprising database tables, procedure results from logic calls and status/error messages;

translate application requests from the application into a delimited second data structure stored in a memory, the second data structure comprising an element for each of the application requests, the application requests being generated in response to user actions in a graphical user interface; [~~and~~]

generate a stream of text-based data in an Extensible Markup Language (XML) format from the second data structure;

transmit the stream;

receive a response; and

change the value of the control based on the response.

27. (Previously presented) A computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to:

receive a stream of text-based data in an Extensible Markup Language (XML) format;

parse the stream into request statements; and

intercept the request statements prior to execution and apply additional logic based on a type or content of the request .

Claims 28-34 (Canceled).